**CAPP 30254: Machine Learning for Public Policy**

**Assignment 3: Pipeline Improvement and Empirical Comparisons**

Ye Ji

May 4th, 2017

**iPython Notebook:** <https://github.com/jiyecapp/machine_learning_works/tree/master/hw3>

**I. Summary**

In this assignment, I ran several classifiers on the credit data from last assignment and compared them with different metrics to look for evidence for recommending what feature, classifiers to use for detecting the likelihood of serious delinquency from a person's credit data.

**II. Goal**

The goal is to improve the pipeline based on the feedback from previous assignments, and add a few components based on what we've covered in the past few lectures.

**Report**

Write a short report (1-2 pages) that compares the performance of the different classifiers across all the metrics for the data set used in the last assignment. Which classifier does better on which metrics? Which one runs faster? What would be your recommendation to someone who's working on the credit model?

**III. Pipeline Improvement**

Below are the five main changes in my pipeline. The first two are made specifically based on TA feedback.

1. Create ways to fill all missing values for discrete/categorical variables.
2. Style improvement to group functions by their purpose.

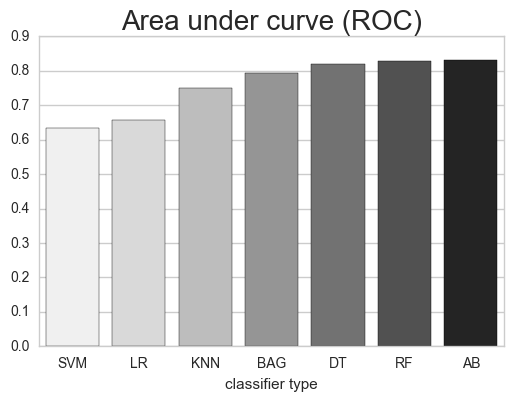
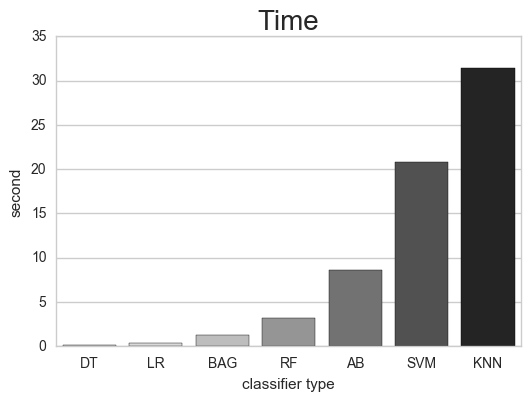
*Note: I have abstract the functions for each step of the pipeline into different files for future use. However, in my iPython notebook, these functions are not imported from the separated files because all the style improvement is done after running the magic loop. I would add the new function to call on the pipeline in the future, but given the computational cost of the magic loop and the time constraint, this main function is not in the repository for now.*

1. A more thorough data exploration, with the addition of plotting the distributions of different variables (with and without outliers), and plotting a heat map of correlation between variables.
2. Additional functions in pre-processing and feature generation.
3. Added Scaling(robust), feature selection(ExtraTree), and test-train split operations.

**IV. Classifier Comparison**

I have used DecisionTree, LogRegression, Bagging, RandomForest, AdaBoosting, SVM(LinearSVC), and K-Nearest Neighbor for this assignment. The following paragraphs will discuss the differences in time, AUC-ROC, accuracy, and AUC-precision/recall of all the classifiers.

***Time:*** In terms of time, KNN is the most expensive (31.48 seconds), followed by SVM (20.78 seconds). Do note that the SVM uses the specification of 'LinearSVC', which reduces the computational cost. If using 'SVC(kernel='linear')', its time would have been considerably longer and become the most computationally expensive classifier instead. (See graph below on the left)

****

***Area Under Curve (ROC):*** In term of area under curve(AUC) of ROC curve, AdaBoost, RandomForest, and DecisionTree is the top three performers. (See graph above on the right)

***F1:*** In term of f1 score, the result is the same as AUC(ROC).

***Accuracy*:** All classifiers perform equivalently on accuracy at 0.93, regardless of parameters. We knew this already from the last assignment. Therefore, accuracy is probably not an appropriate metric for detecting serious delinquency from this credit data.

**V. Implication**

***Feature:*** Using ExtraTree classifiers can help determine the most effective features based on relative feature importance for detecting future delinquency. In this case, revolving utilization of unsecured lines, age, normalized debt ratio, normalized monthly income, and number of open credit lines and loans are selected as the six most relevant features for the prediction. Since some of these features are recorded by credit agencies and even included in public survey. It would be interesting to cross-reference American Community Survey or other public record to detect delinquency in wider geographic area.

***Metric:*** We already know that accuracy performance does not differ among all the classifiers. As for the rest, depending on the identity of the stakeholders, this type of prediction could focus on different metrics. For example, if this is an internal prediction of a credit agency or financial institute, precision is probably more meaningful because identifying true positive in serious delinquency helps the stakeholders to minimize loss.

***Classifier:*** As for the classifiers, KNN and SVM are computational expensive on data of this size. AdaBoost has the best overall perfomances, especially considering the trade-off between effectiveness and time cost. It would be the recommended classifier in this limited study.